



TITLE:

On the NP-Completeness of weakly stable marriage

AUTHOR(S):

Miyazaki, Shuichi; Iwama, Kazuo

CITATION:

Miyazaki, Shuichi ...[et al]. On the NP-Completeness of weakly stable marriage. 電子情報通信学会技術研究報告 1998, 98(432): 33-40: COMP98-55.

ISSUE DATE:

1998-11-20

URL:

<http://hdl.handle.net/2433/227134>

RIGHT:

© 1998 by the Institute of Electronics, Information and Communications Engineers (IEICE)

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
COMP98-55 (1998-11)

条件を緩和した安定結婚問題の NP 完全性について

宮崎 修一 岩間 一雄

京都大学 大学院情報学研究科
〒 606-8501 京都市左京区吉田本町

Tel 075-753-5392

Fax 075-753-5379

{shuichi,iwama}@kuis.kyoto-u.ac.jp

あらまし

安定結婚問題は、入力として N 人ずつの男女と各個人の異性に対する希望リスト（異性 N 人の順序付け）が与えられ、安定な N 組のペアを求める（安定な N 組のペアが存在するかどうかを問う）問題である。本問題の自然な拡張として、(i) 希望リストに異性 N 人全員を書かなくても良いもの、(ii) 希望の順序付けに半順序を許すもの、があるが、いずれも解の存在は多項式時間で判定できることが知られている。本研究では、(i) および (ii) を同時に許した場合に、本問題が NP 完全になることを示す。また、本問題に関連した問題の NP 完全性も示す。

キーワード 安定結婚問題, 半順序, 学生配属問題, NP 完全性

On the NP-Completeness of Weakly Stable Marriage

Shuichi Miyazaki Kazuo Iwama

Graduate School of Informatics, Kyoto University

Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501

Phone +81-75-753-5392

Fax +81-75-753-5379

{shuichi,iwama}@kuis.kyoto-u.ac.jp

Abstract

The original stable marriage problem requires both each man and woman to submit a complete and strict order of his/her preference, i.e., a total order list of, say, 100 people, if there are 100 men and women. This is obviously unrealistic often in practice, and several relaxations have been proposed, including the following two common ones: One is to allow an incomplete list, i.e., a man is allowed to accept only a subset of the women and vice versa. The other is to allow a "weaker" preference list such as a list including ties and partial orders. A little surprisingly, it is known that both problems can still be solved in polynomial time. In this paper, we show that the problem becomes NP-hard, if we allow both relaxations (incomplete lists and partial orders) at the same time.

key words stable marriage problem, partial order, student assignment problem, NP-completeness

1 Introduction

An instance of the *stable marriage problem* [GI89] consists of N men and N women. Each person has his/her *strictly* ordered preference list containing *all* the members of the opposite sex. A matching M is one-one correspondence between the men and the women. If man m and woman w are matched in M , we say m and w are partners in M , and we write $p_M(m) = w$ and $p_M(w) = m$. A man m and a woman w is called a blocking pair for matching M , if m and w are not partners in M , but m prefers w to $p_M(m)$ and w prefers m to $p_M(w)$. If there is no blocking pair for M , then M is called *stable*. If a matching M contains all the $2N$ members, it is called *complete*. Stable marriage problem was first studied by Gale and Shapley [GS62]. They showed that there always exists a complete stable matching in any instance and showed $O(n^2)$ time algorithm (so-called Gale-Shapley algorithm) to find one.

However, considering practical applications, the restriction for preference lists appears to be too strict in many occasions because of the *total order* and *complete* list of preferences. In this context, there are some extensions of this problem. The popular ones are (i) the stable marriage problem with unacceptable partners [GI89], and (ii) the stable marriage problem with indifference [GI89, Ir94]. In the first extension, each person is allowed to declare one or more unacceptable partners. Thus each person's preference list may be incomplete. Gale and Sotomayor [GS85] studied this extension and showed that a maximum stable matching is found in polynomial time using Gale-Shapley algorithm slightly modified to treat the unacceptable case (see also [GI89]). Thus the problem does not become essentially harder.

In the second extension, each person is allowed to have a preference list with ties, or more generally, a preference list of partial order. m and w are called a blocking pair if m and w are not partners in M , but m *strictly* prefers w to $p_M(m)$ and w *strictly* prefers m to $p_M(w)$ in the partial order list. A matching containing no such blocking pair is called *weakly stable*. However, this extension also does not make the problem significantly difficult; it is known that there is a polynomial-time algorithm which determines whether there exists a complete stable matching (and finds one if exists) [GI89, Ir94]. In [Ir94], two other notions of stabilities are also introduced but the situation is still the same, i.e., the problem is solvable in polynomial time.

In this paper, we consider how the situation changes if we apply both extensions, i.e., incomplete preference lists and partial orders, at the same time. Our main purpose of this paper is to show that the situation does change, i.e., the problem is now NP-complete. Also, it is shown that the problem is also intractable for *complete* lists if the question is not the existence of complete matching but obtaining *optimal* (complete) matchings. The general perception on the stable marriage problem has been that it is basically not hard [GS62, GI89, Ir94]. (Two exceptions are mentioned in [Ro90], either of which is not the bipartite setting.) Our results in this paper could give some change to this common perception.

In this paper, we introduce the *student assignment problem*, which is a special case of the marriage problem. There are N graduate students and N professors, where each student has a preference list of professors and students are ordered according to the scores of the entrance examination. We are asked to assign each student to each professor under some stable condition. In Sec.2, we give formal definitions of several extensions of the stable marriage problem and student assignment problem, and show the relationship between them. Sec.3 is devoted for the proof of several intractability results of the marriage problem with unacceptable partners and partial order preference lists. The main proof is given against the student assignment problem.

2 Stable Marriage and Student Assignment Problems

Recall that the original stable matching problem requires each person's preference list to be *complete* (i.e., all the members of the opposite sex must be included) and to be a *total order*. We focus on the three possible relaxations concerning these two restrictions. All these problems (including the

original one) ask if there exists a complete stable matching. In each problem, we are given N men, N women and each person's preference list. The condition for a pair of a man m and a woman w to be a blocking pair is the same for all: m and w is a blocking pair for matching M , if m and w are not partners in M , but m strictly prefers w to $p_M(m)$ and w strictly prefers m to $p_M(w)$. Thus to summarize these problems, it suffices to specify conditions for preference lists:

Problem: Stable Marriage Problem with Complete List and Total Order (SMP-CLTO).

Preference List: Each person's preference list must be complete and must be a total order.

Problem: Stable Marriage Problem with Incomplete List and Total Order (SMP-ILTO).

Preference List: Each person's preference list may be incomplete but must be a total order.

Problem: Stable Marriage Problem with Complete List and Partial Order (SMP-CLPO).

Preference List: Each person's preference list must be complete but may be a partial order.

Problem: Stable Marriage Problem with Incomplete List and Partial Order (SMP-ILPO).

Preference List: Each person's preference list may be incomplete and may be a partial order.

The first three problems are known to be in P [GI89, Ir94]. We later show the NP-completeness of the fourth one. If we consider a *cost* of a matching, we can consider another extension. Let m be a man and $L = (w_1, w_2, \dots, w_N)$ be his preference list. A *weighted* preference list is a combination of L and a weight function σ which maps each w_i into a real number such that if w_i precedes w_j in m 's list, then $\sigma_m(w_i) \leq \sigma_m(w_j)$. Namely, the cost function must not contradict to the total order. If m is matched to w_i in a matching M , then m 's cost under M is defined to be $\sigma_m(w_i)$. A similar cost function is also introduced for the women's preference lists. A cost of the matching M is the sum of the costs of all $2N$ persons.

Problem: k -SMP-CLTO.

Instance: N men, N women, a positive integer k , each person's preference list which must be complete and must be a total order, and a cost function σ .

Question: Is there a complete stable matching whose cost is equal to or smaller than k ?

Problem: k -SMP-CLPO.

Instance: N men, N women, a positive integer k , each person's preference list which must be complete but may be a partial order, and a cost function σ which does not contradict to the partial order.

Question: Is there a complete stable matching whose cost is equal to or smaller than k ?

The first optimization problem is known to be in P [GI89], i.e., a stable matching with minimum cost can be obtained in polynomial time. We later show the NP-completeness of the second problem.

Finally let us consider the *Student Assignment Problem* (SAP for short) introduced in Sec.1. An instance of SAP consists of the same number N of students and professors, the *rank list* of students, and a *preference list* of each student. The rank list is a total order of students. If student s_i is in the j th position on the rank list, we write $r(s_i) = j$. Each student has a preference list of a total order of (not necessarily all) professors. If professor a_j is in the l th position on s_i 's list, we write $\delta_{s_i}(a_j) = l$. An *assignment* A is a bijection from students to professors; if student s_i is assigned to professor a_l , we write $A(s_i) = a_l$. Each s_i must be mapped to a professor who exists on s_i 's preference list. Suppose that, under an assignment A , students s_i and s_j are assigned to professors a_l and a_m , respectively, or $A(s_i) = a_l$ and $A(s_j) = a_m$. Then a pair of a student s_i and a professor a_m is called a *blocking pair* for A if (1) $r(s_i) < r(s_j)$, (2) $\delta_{s_i}(a_m) < \delta_{s_i}(a_l)$ and (3) $\delta_{s_i}(a_m) \leq \delta_{s_j}(a_m)$. If there is no blocking pair for A , then A is called *stable*. Here we consider only one version, i.e., the preference lists of students may be incomplete but must be a total order.

Problem: Student Assignment Problem (SAP).

Instance: N students, N professors, and each student's preference list which may be incomplete but must be a total order.

Question: Is there a stable assignment?

Lemma 1. SAP is a special case of SMP-ILPO.

Proof. We consider each professor's preference list as a partial order of students. First, professor a_l 's preference list does not include student s_i if a_l does not appear on s_i 's preference list. Then consider two students s_i and s_j included in a_l 's list. a_l *strictly* prefers s_i to s_j if and only if (i) $r(s_i) < r(s_j)$, and (ii) $\delta_{s_i}(a_l) \leq \delta_{s_j}(a_l)$. (Blocking conditions (1) and (3) are combined into one, and reflected in a preference lists of professors.) It is not hard to see that the preference of each professor is a partial order of students, namely. A blocking pair then is a pair of a student and a professor, who are not currently matched but strictly prefer each other to their current partners. \square

3 Intractability Results

We first prove that SAP is NP-complete. It immediately implies that SMP-ILPO is NP-complete by Lemma 1. Also it is straightforward to imply that k -SMP-CLPO is NP-hard using some artificial weight function. We also discuss more natural weight functions.

Theorem 1. SAP is NP-complete.

Proof. It is easy to see that SAP is in NP: Given an assignment A , one can check whether A is a bijection or not. Then check, for each pair of a student and a professor, whether it is a blocking pair. This can be done in polynomial time.

To show the NP-hardness, let us consider the following problem:

Problem: ONE-IN-THREE 3SAT.

Instance: 3CNF formula.

Question: Is there any truth assignment such that exactly one literal in each clause is true?

It is known that ONE-IN-THREE 3SAT remains NP-complete even if a 3CNF formula does not include negative literals [GJ79, Sc78]. So, we translate this restricted problem into SAP. Given f , which is an instance of ONE-IN-THREE 3SAT, we construct an instance $T(f)$ of SAP, namely, (1) the same number of students and professors, (2) a rank list of students, and (3) each student's preference list. Let n and m be the numbers of variables and clauses of $f = C_1 \cdot C_2 \cdots C_m$, respectively. Let t_i be the number of appearances of x_i and t be the maximum number among t_1, t_2, \dots, t_n . $T(f)$ consists of $9m + 3n + t + 6$ students and the same number of professors. We first introduce the following $9m + 3n + t + 6$ students:

Group (i): $t + 6$ students $s_{1,1}, \dots, s_{1,t+6}$.

Group (ii): n students $s_{2,1}, \dots, s_{2,n}$.

Group (iii): $2n$ students $s_{3,1}^+, s_{3,1}^-, \dots, s_{3,n}^+, s_{3,n}^-$.

Group (iv): $6m$ students $s_{4,i,j}^+, s_{4,i,j}^-$. For $1 \leq i \leq n$ and $1 \leq j \leq m$, students $s_{4,i,j}^+$ and $s_{4,i,j}^-$ exist if and only if clause C_j contains literal x_i . (Recall that \bar{x}_i never appears in f .) Since there are $3m$ literals, $6m$ students will be introduced.

Group (v): $3m$ students $s_{5,i,j}$ for any $1 \leq i \leq m$ and $1 \leq j \leq 3$. Students $s_{5,j,1}$, $s_{5,j,2}$ and $s_{5,j,3}$ correspond to clause C_j .

Students of Group (i) are necessary for technical reason. A student $s_{2,i}$ of Group (ii) and two students $s_{3,i}^+$ and $s_{3,i}^-$ of Group (iii) are associated with variable x_i . Students $s_{4,i,j}^+$ and $s_{4,i,j}^-$ of Group (iv) are associated with literal x_i in clause C_j . The same number $(9m + 3n + t + 6)$ of professors are divided into the following three categories:

(1) $t + 6$ professors $g_{1,1}, \dots, g_{1,t+6}$.

(2) $3n$ professors i, \bar{i} and P_i ($1 \leq i \leq n$). Three professors i, \bar{i} and P_i are associated with variable x_i .

(3) $9m$ professors (i, j) , $(\overline{i}, \overline{j})$ and $Q_{i,j}$. For $1 \leq i \leq n$ and $1 \leq j \leq m$, three professors (i, j) , $(\overline{i}, \overline{j})$ and $Q_{i,j}$ are prepared if and only if clause C_j contains literal x_i . Since there are $3m$ literals, $9m$ professors will be introduced.

Secondly, we determine the rank of students as follows: Among the groups, the order goes as Group (i) highest, Group (ii) next, and so on, until Group (v) that is lowest. The order within Groups (i), (ii) and (iii) are determined as explicitly described above, e.g., $r(s_{1,1}) < r(s_{1,2})$. Within Group (iv), a student with smaller index i is higher in the rank. If two students have the same value of i , then the student having smaller j is higher. If both i and j are the same, the student with $+$ is higher than the student with $-$. Within Group (v), a student with smaller i is higher. If two students have the same value of i , then the student with smaller j is higher.

Finally, we construct each student's preference list. For better exposition, we use an example of f , i.e., $f_0 = (x_1 + x_2 + x_3)(x_1 + x_3 + x_4)$, for which the preference list is turned out to be as illustrated in Table 2. As for Group-(i) students, each student selects only one professor, i.e., $s_{1,i}$ selects $g_{1,i}$. Students of Group (ii) selects $t + 8$ professors in 1st to $(t + 8)$ th positions of the preference lists. Professors on the 1st through $(t + 6)$ th positions are $g_{1,1}$ through $g_{1,t+6}$, in an arbitrary order. $s_{2,i}$ selects professor i in the $(t + 7)$ th position and selects professor \overline{i} in the $(t + 8)$ th position, namely, $\delta_{s_{2,i}}(i) = t + 7$ and $\delta_{s_{2,i}}(\overline{i}) = t + 8$.

Then we construct preference lists of Group-(iii) students. We show how to construct preference lists using students $s_{3,1}^+$ and $s_{3,1}^-$ for f_0 . (Again, see Table 2.) Recall that these two students are associated with variable x_1 . Students $s_{3,1}^+$ and $s_{3,1}^-$ write professor P_1 in the 5th and the 4th positions, respectively. (The 5th and 4th positions are always used without depending on f .) Then, $s_{3,1}^+$ writes professor 1 in the $t + 6 (= 8)$ th position and $s_{3,1}^-$ writes $\overline{1}$ in the same position. Since x_1 appears in clauses C_1 and C_2 , four professors $(1, 1)$, $(\overline{1}, \overline{1})$, $(1, 2)$ and $(\overline{1}, \overline{2})$ have been introduced. $s_{3,1}^+$ writes $(1, 1)$ and $(1, 2)$ in the 6th and 7th positions, respectively, and $s_{3,1}^-$ writes $(\overline{1}, \overline{1})$ and $(\overline{1}, \overline{2})$ also in the 6th and 7th positions, respectively. Other positions are filled with $g_{1,1}$ through $g_{1,8}$ arbitrarily.

Now we move to Group (iv)-students: We describe how to construct preference lists of $s_{4,1,1}^+$ and $s_{4,1,1}^-$ of Group (iv) in the example, who are associated with variable x_1 . Student $s_{4,1,1}^+$ writes professor $Q_{1,1}$ in the 5th position and writes $(\overline{1}, \overline{1})$ in the same position as $s_{3,1}^+$ wrote it. $s_{4,1,1}^-$ writes $Q_{1,1}$ in the 4th position and writes $(1, 1)$ in the same position as $s_{3,1}^-$ wrote it. Blanks are filled with professors $g_{1,1}$ through $g_{1,8}$ arbitrarily.

Recall that three students $s_{5,j,1}$, $s_{5,j,2}$ and $s_{5,j,3}$ of Group (v) are associated with clause C_j . We show how to construct preference lists of students $s_{5,2,1}$, $s_{5,2,2}$ and $s_{5,2,3}$ associated with $C_2 = (x_1 + x_3 + x_4)$ of f_0 . Since literals x_1 , x_3 and x_4 appear in C_2 , six professors $(1, 2)$, $(\overline{1}, \overline{2})$, $(3, 2)$, $(\overline{3}, \overline{2})$, $(4, 2)$ and $(\overline{4}, \overline{2})$ have been introduced. $s_{5,2,1}$ writes $(1, 2)$, $(\overline{3}, \overline{2})$ and $(\overline{4}, \overline{2})$ in this order from the 1st to the 3rd positions. Both $s_{5,2,2}$ and $s_{5,2,3}$ write $(1, 2)$, $(3, 2)$ and $(4, 2)$ in this order.

Now the translation is completed. Table 2 shows the whole lists of $T(f_0)$. Next, we show a series of lemmas which summarize several conditions for an assignment A for $T(f)$ to be a solution of the problem. We omit the proofs but the correctness can be easily seen. Recall that we denote $A(s_i) = a_j$ if student s_i is assigned to professor a_j under the assignment A .

Lemma 2. If an assignment A for $T(f)$ is a solution, then $A(s_{1,i}) = g_{1,i}$ ($1 \leq i \leq t + 6$). Namely, each student of Group (i) is assigned to the professor of his/her first position on the list.

Lemma 3. If an assignment A for $T(f)$ is a solution, then for each i ($1 \leq i \leq n$), $A(s_{2,i}) = i$ or $A(s_{2,i}) = \overline{i}$.

Lemma 4. Suppose that an assignment A for $T(f)$ is a solution. Then for all $1 \leq i \leq n$, (a) if $A(s_{2,i}) = i$ then $A(s_{3,i}^+) = P_i$ and $A(s_{3,i}^-) = \overline{i}$. (b) Otherwise, i.e., if $A(s_{2,i}) = \overline{i}$, then $A(s_{3,i}^+) = i$ and $A(s_{3,i}^-) = P_i$.

Lemma 5. Suppose that an assignment A for $T(f)$ is a solution. Then following (a) and (b) hold for all i ($1 \leq i \leq n$). (a) If $A(s_{2,i}) = i$ then $A(s_{4,i,j}^+) = (\overline{i}, \overline{j})$ and $A(s_{4,i,j}^-) = Q_{i,j}$ for all j . (b) If $A(s_{2,i}) = \overline{i}$ then $A(s_{4,i,j}^+) = Q_{i,j}$ and $A(s_{4,i,j}^-) = (i, j)$ for all j .

Lemma 6. Suppose that an assignment A for $T(f)$ is a solution. Then, for each i and j , one

of (i, j) and (\bar{i}, j) is assigned to a student of Group (iv), and the other is assigned to a student of Group (v).

Let $C_j = (x_{j_1} + x_{j_2} + x_{j_3})$ be a clause in f , where $j_1 < j_2 < j_3$. Then recall that there are three students $s_{5,j,1}$, $s_{5,j,2}$ and $s_{5,j,3}$, and six professors (j_1, j) , (\bar{j}_1, j) , (j_2, j) , (\bar{j}_2, j) , (j_3, j) and (\bar{j}_3, j) (see Table 1).

$s_{5,j,1}$	(\bar{j}_1, j)	(j_2, j)	(\bar{j}_3, j)						
$s_{5,j,2}$	(j_1, j)	(j_2, j)	(j_3, j)						
$s_{5,j,3}$	(j_1, j)	(\bar{j}_2, j)	(j_3, j)						

Table 1: Preference lists of students associated to C_j

Lemma 7. Suppose that an assignment A for $T(f)$ is a solution. Then, for all j ($1 \leq j \leq m$), $s_{5,j,1}$, $s_{5,j,2}$ and $s_{5,j,3}$ are assigned in one of the following ways (a), (b) and (c): (a) $A(s_{5,j,1}) = (\bar{j}_1, j)$, $A(s_{5,j,2}) = (j_2, j)$ and $A(s_{5,j,3}) = (j_3, j)$. (b) $A(s_{5,j,1}) = (\bar{j}_2, j)$, $A(s_{5,j,1}) = (j_1, j)$ and $A(s_{5,j,3}) = (j_3, j)$. (c) $A(s_{5,j,1}) = (\bar{j}_3, j)$, $A(s_{5,j,2}) = (j_1, j)$ and $A(s_{5,j,3}) = (j_2, j)$.

Lemma 8. Suppose that an assignment A for $T(f)$ is a solution. Then, for each i and j , the following statements (a), (b) and (c) are true: (a) If $A(s_{5,j,1}) = (\bar{i}, j)$ then $A(s_{2,i}) = \bar{i}$. (b) If $A(s_{5,j,2}) = (i, j)$ then $A(s_{2,i}) = i$. (c) If $A(s_{5,j,3}) = (i, j)$ then $A(s_{2,i}) = i$.

Now we are ready to show the correctness of the reduction. To make the argument clear, we denote the literal x_i in clause C_j by x_i^j . Let us consider the following association rule between assignment for variables (and literals) of f and assignment A for students of $T(f)$: (1) Assign 1 to *variable* x_i if and only if student $s_{2,i}$ is assigned to professor \bar{i} , and assign 0 to *variable* x_i if and only if student $s_{2,i}$ is assigned to professor i . (2) Assign 1 to *literal* x_i^j if and only if student $s_{5,j,1}$ is assigned to professor (\bar{i}, j) , and assign 0 to *literal* x_i^j if and only if student $s_{5,j,2}$ or $s_{5,j,3}$ is assigned to professor (i, j) . We will show the consistency, namely, under the assumption that A is a solution for $T(f)$, if $x_i = 1$ then $x_i^j = 1$ for all j , and if $x_i = 0$ then $x_i^j = 0$ for all j .

Suppose $x_i = 1$. Then $A(s_{2,i}) = \bar{i}$ by the association rule. Suppose, for some j such that literal x_i exists in clause C_j , $s_{5,j,2}$ or $s_{5,j,3}$ is assigned to professor (i, j) . Then, by (b) or (c) of Lemma 8, $A(s_{2,i}) = i$, which is a contradiction, since $s_{2,i}$ is assigned to both i and \bar{i} under A . Thus if $x_i = 1$ then $x_i^j = 1$ for all j . The other one, i.e., for $x_i = 0$, can be shown similarly.

Suppose there exists a solution A^* for $T(f)$. By Lemma 3, either $A^*(s_{2,i}) = i$ or $A^*(s_{2,i}) = \bar{i}$ for all i . Then we determine the assignment to variables of f using the association rule: If $A^*(s_{2,i}) = i$ then $x_i = 0$, otherwise, i.e., if $A^*(s_{2,i}) = \bar{i}$, then $x_i = 1$. We show that this is a solution for f . Let, for $1 \leq j \leq m$, the j th clause of f be $C_j = (x_{j_1} + x_{j_2} + x_{j_3})$. Then preference lists of three students associated to C_j are the ones described in Table 1. By Lemma 7, these students must be assigned in either (a) or (b) or (c) in Lemma 7. Suppose it is (a), namely, $A^*(s_{5,j,1}) = (\bar{j}_1, j)$, $A^*(s_{5,j,2}) = (j_2, j)$ and $A^*(s_{5,j,3}) = (j_3, j)$. Then, by the association rule, we have $x_{j_1}^j = 1$, $x_{j_2}^j = 0$ and $x_{j_3}^j = 0$ for *literals*, namely, exactly one literal is true. (It is not hard to see that, in both cases (b) and (c) also, exactly one literal in C_j is true.) Since A^* is a solution for $T(f)$, the above consistency says that there is no contradiction between the value of each *variable* and the value of each *literal*. Thus this assignment is a solution for f .

Conversely, suppose that there exists a solution for f of ONE-IN-THREE 3SAT. Then, again using the association rule, we determine assignment for students of Groups (ii) and (v). Students of Groups (i), (iii) and (iv) are automatically assigned. The fact that this assignment is a solution for $T(f)$ can be easily seen from the lemmas we have already shown. \square

Corollary 1. SMP-ILPO is NP-complete.

Proof. Membership in NP is obvious. We have shown, in Lemma 1, that SMP-ILPO is a special case of SAP. \square

Corollary 2. k -SMP-CLPO is NP-complete.

Proof. Membership in NP is obvious. We will give a reduction from SMP-ILPO. We first make each man m 's preference list complete: We add a woman w (who did not exist in m 's original list) to m 's list in such a way that m does not prefer w to any other women who did originally exist in the list. We define a weight function in the following way: The weight of each woman who was in m 's original list is 0, and the weight of each woman newly added is 1. We do the similar process to each woman's list also. Then it is easily seen that there exists a complete stable matching for SMP-ILPO if and only if there exists a stable matching for k -SMP-CLPO whose cost is 0. \square

Let us consider a much more natural cost function, denoted by C , than the one used in Corollary 2. Define s_i 's cost under A be $C_A(s_i) = \delta_{s_i}(A(s_i))$, namely, the position of the professor whom s_i is assigned to. (δ is defined in Sec.2) The cost of an assignment A is the sum of the costs of all the students under A . As one can see easily, this cost function can be used for SMP-ILPO if the preference lists of either men or women are total order.

Problem: k -SAP.

Instance: N students, N professors, a positive integer k , and each student's preference list which must be complete and must be a total order.

Question: Is there a complete stable assignment whose cost is equal to or smaller than k ?

Theorem 2. k -SAP is NP-complete.

Proof (Sketch). Membership in NP is obvious. We reduce $T(f)$ obtained in the proof of Theorem 1 into an instance $T'(T(f))$ of k -SAP. We introduce a large number, say n^2m^2 , of Group-(i) students $s_{1,j}$ and professors $g_{1,j}$ and make each student's preference lists complete by first adding $g_{1,j}$'s to the tail of the list and then other remaining professors. It can be seen that there is no complete matching in $T(f)$ if and only if at least one student in $T'(T(f))$ is assigned to a professor of very low position on the list. If we set appropriate k , only the cost of this student exceeds k . \square

References

- [GS62] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, Vol.69, pp.9-15, 1962.
- [GS85] D. Gale and M. Sotomayor, "Some remarks on the stable matching problem," *Discrete Applied Mathematics*, Vol.11, pp.223-232, 1985.
- [GJ79] M. R. Garey and D. S. Johnson, "Computers and Intractability, A Guide to The Theory of NP-Completeness," Freeman, San Francisco, 1979.
- [GI89] D. Gusfield and R. W. Irving, "The Stable Marriage Problems: Structure and Algorithms," MIT Press, Boston, MA, 1989.
- [Ir85] R. W. Irving, "An efficient algorithm for the stable room-mates problem," *J. Algorithms*, Vol.6, pp.577-595, 1985.
- [Ir94] R. W. Irving, "Stable marriage and indifference," *Discrete Applied Mathematics*, Vol.48, pp.261-272, 1994.
- [MMI98] Y. Morita, S. Miyazaki and K. Iwama, "Partial MAXSAT for combinatorial optimization on university databases," *Proc. IPSJ-DBS*, 98-DBS-116(2), pp.335-342, 1998 (in Japanese).
- [Ro90] E. Ronn, "NP-complete stable matching problems," *J. Algorithms*, Vol.11, pp.285-304, 1990.
- [Sc78] T. J. Schaefer, "The complexity of satisfiability problems," *Proc. 10th Symp. on Theory of Computing*, pp.216-226, 1978.

$s_{1,1}$	$g_{1,1}$									
$s_{1,2}$	$g_{1,2}$									
$s_{1,3}$	$g_{1,3}$									
$s_{1,4}$	$g_{1,4}$									
$s_{1,5}$	$g_{1,5}$									
$s_{1,6}$	$g_{1,6}$									
$s_{1,7}$	$g_{1,7}$									
$s_{1,8}$	$g_{1,8}$									
$s_{2,1}$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$g_{1,5}$	$g_{1,6}$	$g_{1,7}$	$g_{1,8}$	1	$\bar{1}$
$s_{2,2}$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$g_{1,5}$	$g_{1,6}$	$g_{1,7}$	$g_{1,8}$	2	$\bar{2}$
$s_{2,3}$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$g_{1,5}$	$g_{1,6}$	$g_{1,7}$	$g_{1,8}$	3	$\bar{3}$
$s_{2,4}$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$g_{1,5}$	$g_{1,6}$	$g_{1,7}$	$g_{1,8}$	4	$\bar{4}$
$s_{3,1}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	P_1	$(1,1)$	$(1,2)$	1		
$s_{3,1}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	P_1	$g_{1,5}$	$(1,1)$	$(1,2)$	$\bar{1}$		
$s_{3,2}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	P_2	$(2,1)$	$g_{1,7}$	2		
$s_{3,2}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	P_2	$g_{1,5}$	$(2,1)$	$g_{1,7}$	$\bar{2}$		
$s_{3,3}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	P_3	$(3,1)$	$(3,2)$	3		
$s_{3,3}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	P_3	$g_{1,5}$	$(3,1)$	$(3,2)$	$\bar{3}$		
$s_{3,4}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	P_4	$(4,2)$	$g_{1,7}$	4		
$s_{3,4}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	P_4	$g_{1,5}$	$(4,2)$	$g_{1,7}$	$\bar{4}$		
$s_{4,1,1}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$Q_{1,1}$	$(1,1)$				
$s_{4,1,1}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$Q_{1,1}$	$g_{1,5}$	$(1,1)$				
$s_{4,1,2}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$Q_{1,2}$	$g_{1,6}$	$(1,2)$			
$s_{4,1,2}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$Q_{1,2}$	$g_{1,5}$	$g_{1,6}$	$(1,2)$			
$s_{4,2,1}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$Q_{2,1}$	$(2,1)$				
$s_{4,2,1}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$Q_{2,1}$	$g_{1,5}$	$(2,1)$				
$s_{4,3,1}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$Q_{3,1}$	$(3,1)$				
$s_{4,3,1}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$Q_{3,1}$	$g_{1,5}$	$(3,1)$				
$s_{4,3,2}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$Q_{3,2}$	$g_{1,6}$	$(3,2)$			
$s_{4,3,2}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$Q_{3,2}$	$g_{1,5}$	$g_{1,6}$	$(3,2)$			
$s_{4,4,2}^+$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$Q_{4,2}$	$(4,2)$				
$s_{4,4,2}^-$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$Q_{4,2}$	$g_{1,5}$	$(4,2)$				
$s_{5,1,1}$	$(1,1)$	$(2,1)$	$(3,1)$							
$s_{5,1,2}$	$(1,1)$	$(2,1)$	$(3,1)$							
$s_{5,1,3}$	$(1,1)$	$(2,1)$	$(3,1)$							
$s_{5,2,1}$	$(1,2)$	$(3,2)$	$(4,2)$							
$s_{5,2,2}$	$(1,2)$	$(3,2)$	$(4,2)$							
$s_{5,2,3}$	$(1,2)$	$(3,2)$	$(4,2)$							

Table 2: Preference lists of students of $T(f_0)$